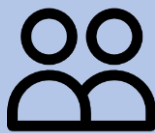


Taller en Sala Nro. 6 ArrayList



En la vida real, el `ArrayList` se utiliza en páginas web para almacenar información que cambia dinámicamente y que podemos saber el número de elementos a priori como los mensajes que recibimos en el correo, las líneas de un chat como Whatsapp, el número de posts en una red social como Twitter o Facebook



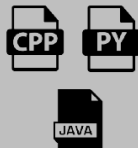
Trabajo en
Parejas



Hoy, plazo
máximo de
entrega



Docente entrega
código suelto en
GitHub



Sí .cpp, .py
o .java



No .zip, .txt,
html o .doc



Alumnos
entregan
código suelto
por GitHub

Ejercicios a resolver

1. Implemente su propia clase `ArrayList`. En particular, un constructor, el método `size`, el método `get` y el método `add`.

2.[Ejercicio Opcional] Implemente un algoritmo que lea una cantidad indeterminada de enteros usando `Scanner` y los guarde en un `ArrayList` de forma invertida, es decir, primero los últimos y luego los primeros.

Al ingresar -1 debe terminar la lectura de los números. Finalmente, imprima el `ArrayList` para verificar que funciona correctamente el algoritmo.

3 [Ejercicio Opcional] Implemente un algoritmo que reciba por parámetro un número $N > 0$ y genere un *Arraylist* con el patrón $\{1, 1, 2, 1, 2, 3, \dots 1, 2, 3, \dots n\}$.

Finalmente, imprima el *Arraylist* para verificar que funcional el algoritmo.

Ejemplos:

N=1 [1]

N=2 [1, 1, 2]

N=4 [1, 1, 2, 1, 2, 3, 1, 2, 3, 4]

Ayudas para resolver los Ejercicios

Ayudas para el Ejercicio 1.....

Pág. 4

Ayudas para el Ejercicio 1



Pista 1: Utilice el método *Arrays.copyOf(...)*



Pista 2: Lance la excepción *IndexOutOfBoundsException* de ser necesario



Pista 3: Diferencien *MiArrayList* de *ArrayList* del API de Java. Un error común es creer que todo se soluciona llamando los métodos existentes en *ArrayList* y, no es así, la idea es implementar una lista con arreglos nosotros mismos. A continuación, un ejemplo del error:

```
// Retorna el tamaño actual de la lista
public int size()
{
    return size();
}

// Retorna el elemento en la posición index
public int get(int index)
{
    return get(index);
}

// Inserta un dato en la posición index
public void insert(int data, int index)
{
    if (index <= size())
    {
        insert(data, index);
    }
}

// Borra el dato en la posición index
public void remove(int index)
{
    remove(index);
}

// Verifica si está un dato en la lista
public boolean contains(int data)
{
    return contains(data);
}
```

```
import java.util.Arrays;
public class MiArrayList {
    private int size;
    private static final int DEFAULT_CAPACITY = 10;
    private int elements[];
```

// Inicializa los atributos *size* en cero y *elements* como un arreglo de tamaño *DEFAULT_CAPACITY*. No, no recibe parámetros.

```
    public MiArrayList() {

    }
```

// Retorna el tamaño de la lista

```
    public int size() {

    }
```

// Agrega un elemento e a la última posición de la lista

```
    public void add(int e) {

    }
```

// Retorna el elemento que se encuentra en la posición i de la lista

```
    public int get(int i) {

    }
```

// Agrega un elemento e en la posición index de la lista

```
    public void add(int index, int e) {

    }
}
```



Como un ejemplo, para el ArrayList [1,2,3,4], el método get(1) retorna el número 2, el método add(2,5) cambia el ArrayList a [1,2,5,3,4], el método add(6) cambiar el ArrayList a [1,2,5,3,4,6] y el método size() retorna 6.

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 **Ext.** 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agende una cita con él a través de **<http://bit.ly/2gzVg10>** , en la pestaña *Semana*. Si no da clic en esta pestaña, parecerá que toda la agenda estará ocupada.